UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/527,812	11/29/2005	Christophe Justin Evrard	550-619	4576

23117 7590 03/18/2008

NIXON & VANDERHYE, PC
901 NORTH GLEBE ROAD, 11TH FLOOR
ARLINGTON, VA 22203

EXAMINER	
VICARY, KEITH E	
ART UNIT	PAPER NUMBER
2183	
MAIL DATE	DELIVERY MODE
03/18/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/527,812
Filing Date: November 29, 2005
Appellant(s): EVRARD ET AL.

Stanley C. Spooner
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 1/28/2008 appealing from the Office action mailed 5/14/2007.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6804782 B1	**Qiu et al.**	**10-2004**
6625737 B1	**Kissell**	**9-2003**

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

2. Claims 1-2, 5-7, and 10 are rejected under 35 U.S.C. 102(e) as being anticipated by Qiu et al. (Qiu) (US 6804782 B1).

3. **Consider claim 1**, Qiu discloses an apparatus for processing data, comprising a processor core (col. 4, lines 54-55, processor) operable to execute data processing instructions to generate result data values (col. 3, lines 24-28; mathematical operations); and

data processing registers holding data values defining state of said processor core to which said result data values are written (col. 7, lines 19-21, registers; more specifically, Figure 8, memory allocation column, x register and e register); wherein

at least one data processing instruction executed by said processor core is a conditional write data processing instruction (col. 3, lines 50-56 and col. 4, lines 1-5; the multiplication operations; the storing of the result is conditional based on the private key) encoding condition codes specifying conditions under which said conditional write data processing instruction will or will not be permitted to write data to effect a change in state of said processor core (col. 4, lines 23-32 and 61-63, col. 5, lines 22-32; the cryptographic key determines whether the multiplication operation is emulated or not, and this cryptographic key is checked upon determining that an operation is encoded as a multiplication operation; note that the arguments section below explains the instruction encoding condition codes): and further comprising

a trash register to which a result data value may be written instead of a data processing register upon execution of said conditional write data processing instruction when said condition codes within said conditional write data processing instruction do

not permit a write to effect a change in state of said processor core (col. 3, lines 28-31, store to memory that is unnecessary and lines 61-65, always performed the multiplication regardless of the value of the bit, col. 4, lines 1-11, 23-35 and col. 5, lines 37-47; the second memory correlates to the trash register).

4. **Consider claim 6**, Qiu discloses a method of processing data, comprising generating result data values upon execution by a processor core of data processing instructions (col. 4, lines 54-55, processor, and col. 3, lines 24-28; mathematical operations), at least one data processing instruction executed being a conditional write data processing instruction (col. 3, lines 50-56 and col. 4, lines 1-5; the multiplication operations; the storing of the result is conditional based on the private key) encoding condition codes specifying conditions under which said conditional write data processing instruction will or will not be permitted to write data to effect a change in state of said processor core (col. 4, lines 23-32 and 61-63, col. 5, lines 22-32; the cryptographic key determines whether the multiplication operation is emulated or not, and this cryptographic key is checked upon determining that an operation is encoded as a multiplication operation; note that the arguments section below explains the instruction encoding condition codes) and wherein

a result data value is not written to a data processing register holding a data value defining state of said processor core (col. 6, line 5-9 discloses of writing to a data processing register only if the value of the key bit is “1”, col. 6, lines 9-19 disclose that the result data value is not written to a data processing register when the value of the

key bit is "0"); when condition codes within said condition write data processing instruction do not permit a write to effect a change in state of said processor core but is instead written to a trash register (col. 3, lines 28-31, store to memory that is unnecessary and lines 61-65, always performed the multiplication regardless of the value of the bit, col. 4, lines 1-11, 23-35 and col. 5, lines 37-47, wherein the second memory correlates to the trash register and is only written to when the value of the key bit is “0”).

5. **Consider claims 2 and 7**, Qiu discloses said data processing register is part of a register bank having a plurality of data registers to which result data values are written (col. 5, lines 37-47; together the first memory and the second memory make up one bank in the same overall location as in Figure 4; also note that as in claims 1 and 6 above, the later mentioned registers are one embodiment of these memories).

6. **Consider claims 5 and 10**, Qiu discloses said trash register is part of said register bank (col. 5, lines 37-47; together the first memory and the second memory make up one bank in the same overall location as in Figure 4), said trash register being unmapped to a register number such that said trash register may not be specified by a register specifying operand value (col. 5, lines 37-47; given that the second memory is used exclusively for when unnecessary stores to memory are required, and an unnecessary store is only deemed unnecessary based on the cryptographic key, and not based on any arguments/parameters in the instruction, it is inherent that the second

memory cannot be specifically specified as an operand. Furthermore, in col. 7, lines 21-23 and 43-45, the unnecessary store is stored in a temporary register, which is well-known in the art to mean a register which is not user-addressable but is instead used for intermediate calculations).

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 3-4 and 8-9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Qiu as applied to claims 1 and 6 above, and further in view of Kissell (US 6625737 B1).

9. **Consider claims 3 and 8**, Qiu discloses of programmably disabling writing to said trash register (col. 6, lines 46-52 of disabling writing to said trash register after a certain amount of iterations in order to prevent the total disruption of the process, and an embodiment of this concept using a counter is shown in Figure 8, step 7.

Although Qiu discloses that the contents of a counter is used to programmably disable writing to said trash register, Qiu does not explicitly disclose of a *trash register control signal* which programmably disables writing to said trash register because Qiu does not explicitly disclose of how the value of the counter travels to whatever logic is comparing the value with zero, as seen in step 7 of Figure 8.

On the other hand, Kissell discloses of the concept of a signal (col. 8, lines 17, inhibit/burn signal line).

Rationales for arriving at a conclusion of obviousness suggested by the Supreme Court's decision in KSR include combining prior art elements according to known methods to yield predictable results, and it would have been readily recognized to one of ordinary skill in the art at the time of the invention that *a signal* would be able to be used to carry the value in the counter to whatever logic is comparing the value with zero.

Therefore, a sufficient rationale for arriving at a conclusion of obviousness has been arrived at.

10. **Consider claims 4 and 9**, Qiu discloses that said trash register control signal is stored in a system configuration register (Figure 8, counter_Register).

(10) Response to Argument

11. The overall concept of the instant application as embodied by the independent claim and the prior art used by examiner will first be briefly explained.

Consider a cryptographic system which receives input data, encodes it, and outputs the encoded data, wherein the encoding algorithm contains conditional operations that are dependent on a value of a secret cryptographic key that cannot be directly determined by accessing the memory location in which it is stored. Enemy

attackers can nevertheless deduce the value of the secret cryptographic key by measuring the power consumption of various system components; a sudden spike in power consumption by a certain system component signifies that a certain operation which uses that system component was executed, and if that operation was conditional on a value of a secret cryptographic key, the value of that secret cryptographic key would be known to be a certain value. This system component can be either, or both, of an execution unit such as a multiplier, or a register file.

To prevent this method of attack (generally described as power analysis), the independent claim describes an invention in which a conditional operation writes a result to a trash register instead of a data processing register when a conditional write data processing operation is resolved to not normally write. In writing to a register in general regardless of the outcome of the conditional write data processing operation, an attacker attempting to utilize power analysis would not be able to determine how a conditional write data processing operation in an algorithm was truly resolved because it would always *appear* as if the conditional write data processing operation was resolved to be true (e.g. perform a write), and thus the value of the secret cryptographic key remains safe.

The examiner believes that the Qiu reference teaches the claimed invention as described above.

12. Appellant's arguments will now be refuted in sequential order.

13. Appellant first argues on page 7 of the appeal brief (Argument A) of the general differences between the claimed invention and the Qiu reference. This argument essentially states that while Qiu's device overcomes a simple power analysis (SPA), it can be defeated using differential power analysis (DPA), whereas the presently claimed invention addresses the same problem addressed in Qiu, but not only defeats SPA, but DPA as well. However, this argument is incorrect. It is first noted that both the instant claims and the instant specification do not appear to explicitly disclose of the categories of simple power analysis or differential power analysis at all. Moreover, as will be readily seen by the response to arguments further below, Qiu teaches the limitations of writing a result data value to the trash register when the condition codes do not permit a write to effect a change in the processor core (for an introductory frame of reference, see col. 4, lines 1-11 of Qiu, which discloses that when an unnecessary multiplication is performed, the result can be stored to memory as well…[e]ven though the result that is stored to memory is not required by the algorithm, the power and time usage that results from performing the operation will help deter an attacker from deciphering the value of the decryption key); therefore, any types of power analysis which are overcome by the claimed limitations of the instant application would be overcome by the Qiu reference as well, and the solutions of Qiu and the instant claims are not "dramatically different," or different at all. Although there *may* be aspects of the instant specification which overcome differential power analysis, these aspects are in no way present in the instant claims. Appellant argues that the presently claimed invention recognizes that, even on the level of single instructions being executed by a processor core, there is still

a detectable change in power usage (detectable using DPA) where data is written to a particular register and where data is not written to a register; however, Qiu recognizes this as seen again in col. 4, lines 7-11.

Examiner would like to reiterate that applicant's repeated use of "simple power analysis" and "differential power analysis" are irrelevant as these words are not present in the instant claims and specification, and Appellant does not sufficiently correlate the aforementioned terms to the instant invention or prior art reference aside from concluding without any explanation that Qiu is defeated by differential power analysis while the presently claimed invention is not.

14. Appellant next argues on page 9 of the appeal brief (Argument B) that Qiu does not disclose the claimed trash register. However, Qiu discloses in col. 3, lines 28-31, that "the processes could include a store to memory that is unnecessary, i.e. not required by the normal functioning of the algorithm". This memory is the trash register as further explained. Qiu in col. 6, lines 9-19, discloses that "if the value of the bit in the key is "0", no modular multiplication is necessary; yet, to fool the attacker the normal multiplication/modular reduction/store to memory process is implemented….this value is the modularly reduced by n *and stored in memory location 516 – rather than memory location 512*" (emphasis added by examiner), and in Figure 5 shows that element 516 is a register. Also, the bottom of Figure 10 (explained in col. 7, lines 40-45) show of temp1_reg as the trash register. Therefore, Qiu does disclose the claimed trash register.

15. Appellant continues to argue on page 9 of the appeal brief (Argument C) that Qiu does not disclose a trash register to which a result data value will be written instead of a data processing register. However, Qiu does teach a trash register as cited by the examiner in the previous argument. Moreover, Qiu does teach that a result data value will be written to a trash register instead of a data processing register as cited above. To reiterate, Qiu in col. 6, lines 9-19, discloses that "if the value of the bit in the key is "0", no modular multiplication is necessary; yet, to fool the attacker the normal multiplication/modular reduction/store to memory process is implemented….this value is the modularly reduced by n *and stored in memory location 516 – rather than memory location 512*" (emphasis added by examiner), and in Figure 5 shows that element 516 is a register. Also, the bottom of Figure 10 (explained in col. 7, lines 40-45) show of temp1_reg as the trash register and x register as the data processing register. Therefore, Qiu does disclose the claimed trash register to which a result data value will be written instead of a data processing register.

Moreover, Qiu further discloses of writing the result data value to the trash register "when said condition codes within said conditional-write data processing instruction do not permit a write to effect a change in state of said processor core" in col. 3, lines 50-67, which discloses that whether a multiplication result is used is conditioned on whether the value of a bit of a private key is "0" or "1". Note that, in this case, the multiplication operation dependent on the value of the bit of the private key is the conditional-write data processing instruction, and it contains condition codes in that the

opcode which designates the instruction as a multiplication instruction necessitates that the bit of a private key is retrieved to be used as a condition for execution. In other words, in the system of Qiu, the bits of an instruction which designate that the instruction is a multiplication instruction are the condition codes, in that the presence of those bits in tandem with a private key bit of zero prevents a write from effecting a change in the state of a processor core. This interpretation of condition codes is fully valid and is also in fact supported by the Appellant's specification on page 8, lines 14-17, which states that "[a conditional branch instruction BEQ (branch upon equal)] instruction *encodes the behavior* that the specified branch will be performed if the flag indicating an equal result from previous processing is set and will be suppressed if this flag is not set" (emphasis added by examiner). Just as a BEQ instruction encodes the behavior that the instruction will be performed if a flag is set and not performed if the flag is not set, so too does Qiu's multiplication instruction encode the conditional behavior that the instruction will be performed and affect future processing by changing processor state if a bit of a private key is set and not performed if the bit of a private key is not set. Therefore, Qiu does disclose of writing the result data value to the trash register "when said condition codes within said conditional-write data processing instruction do not permit a write to effect a change in state of said processor core."

16. Appellant argues on page 10 of the appeal brief (Argument D) that Qui fails to disclose any concept of a single instruction being executed in one of two different ways, and cites col. 4, lines 21-35, which states that a bit value in a binary key may cause an

unnecessary mathematical operation that would not be required for a normal application of the algorithm to be performed. However, Qui explicitly discloses of a single instruction being executed in one of two different ways (again, see col. 3, lines 61-65 of Qui, which discloses that the same multiplication with the same circuitry would be used regardless of whether the value of the bit is 0 or 1, with the only difference being where the result is written as per col. 6, lines 17-19).

Appellant also argues that Qiu's solution does not prevent differential power analysis and would thus lead one of ordinary skill in the art away from Appellants' claimed apparatus. However, given that Qiu teaches all the claimed limitations of the instant application, examiner is unsure as to how Appellant's invention *would* prevent differential power analysis. Again, Appellant does not elaborate on this issue, and there is no specific reasoning as to how Qiu would lead one of ordinary skill in the art away from Appellants' claimed apparatus and the claimed interrelationship between apparatus elements.

17. Appellant argues on page 12 of the appeal brief (Argument E) that examiner is attempting to interpret the word "instruction" to cover whole sections of algorithmic procedure. Although this interpretation is entirely plausible, examiner will also explain how this interpretation is not necessary for Qiu to teach the claimed limitations.

Qiu teaches in col. 7, line 27 of a subroutine MonPro that performs a Montgomery Product. Performing a Montgomery Product, and thus the code line including MonPro, would be considered an instruction. While it may be true that the

execution of MonPro necessarily entails the execution of many other instructions, this is no different than a “regular” add instruction necessarily entailing the execution of many other micro-ops, such as “read the operand from Register A and send the operand to ALU input 1,” “read the operand from Register B and send the operand to ALU input 2,” “and so forth. Alternatively, consider a taxi driver who receives a command to drive to a certain building. Although this command is in essence composed of smaller instructions, such as “turn left on street X," and take exit Y on highway Z,” it would have been readily recognized that the command would still be considered an instruction despite being comprised of "sub-instructions." Therefore, a “simple” multiply operation and the Montgomery multiply operation would both be considered instructions. Furthermore, it would have been readily recognized to one of ordinary skill in the art at the time of the invention that the interpretation of the Montgomery multiply operation as an instruction does not change any facet of Qiu's general invention, described in col. 1, lines 48-50 (one embodiment of the invention performs unnecessary mathematical operations and/or unnecessary storage of data in order to disguise whether a mathematical operation or store actually took place in an algorithm). Note that this citation is stating that the mathematical operations are *in* an overall algorithm, not that the mathematical operations *are* an algorithm; it is irrelevant in regard to the overall invention of Qiu as to whether the mathematical operation is a subroutine or not). Therefore, the line of code of if(ei==1) x=MonPro(M,x) else return MonPro(M,x) to Temp1_Reg) can be collectively considered an instruction (the counter-related instructions are described in col. 6, lines 46-52 as optional).

Alternatively, it is noted in the above line of code that MonPro(M,x) is executed *regardless of the value of the bit of the private key*, because it is where the result data value is written to which is what depends on the value of the bit of the private key. In other words, regardless of if the multiplication operation is a subroutine or not, the one aspect of the instruction which is truly conditional is the write destination. Therefore, the conditional-write data processing instruction would be the machine instruction which ultimately writes the result of the MonPro(M,x) subroutine to the appropriate register depending on the value of the bit of the private key.

In fact, both interpretations are valid and, additionally, both interpretations may co-exist, depending on whether the invention of Qiu is being observed from the high-level programming language viewpoint or the machine language viewpoint. Regardless of which interpretation is used, the examiner has explained how Qiu teaches the broad limitation of a "conditional-write data processing instruction."

Appellant argues that the examiner attempts to consider that a 0 occurring in a private key is a condition code which is encoded in an instruction to determine how that instruction is to be executed. Although this has been countered in the response to Argument C, the pertinent aspect of that response to argument will be summarized below.

In the system of Qiu, the multiplication instruction itself comprises the condition codes, in that the presence of the multiplication instruction in tandem with a private key bit of zero prevents a write from effecting a change in the state of a processor core. This interpretation of condition codes is fully valid and is also in fact supported by the

Appellant's specification on page 8, lines 14-17, which states that "[a conditional branch instruction BEQ (branch upon equal)] instruction encodes the behavior that the specified branch will be performed if the flag indicating an equal result from previous processing is set and will be suppressed if this flag is not set." Just as a BEQ instruction encodes the behavior that the instruction will be performed if a flag is set and not performed if the flag is not set, so too does Qiu's multiplication instruction encode the conditional behavior that the instruction will be performed and affect future processing by changing processor state if a bit of a private key is set and not performed if the bit of a private key is not set.

It is noted that while the examiner believes that a 0 occurring in a private key is a condition code (in the same way that it would have been readily recognized to one of ordinary skill in the art at the time of the invention that a zero flag would be considered a condition code), examiner is nevertheless using the multiplication instruction itself as explained above as teaching the entire limitation of' "a data processing instruction encoding condition codes." This interpretation is valid, as validated by the Appellant's specification as cited above.

18. Appellant argues on page 13 of the appeal brief (Argument F) that examiner has failed to provide any evidentiary support for a rejection, and refers back to previously made arguments. However, examiner has provided sufficient evidentiary support for a rejection, as explained above.

19. Appellant argues on page 14 of the appeal brief (Argument G) that Kissel does not teach various limitations; however, as detailed above, Qiu does teach those limitations. Appellant further argues that there is no motivation for combining the Qiu and Kissell references, but does not elaborate. Examiner will summarize this rejection.

Qiu discloses of programmably disabling writing to said trash register (col. 6, lines 46-52 of disabling writing to said trash register after a certain amount of iterations in order to prevent the total disruption of the process, and an embodiment of this concept using a counter is shown in Figure 8, step 7.

Although Qiu discloses that the contents of a counter is used to programmably disable writing to said trash register, Qiu does not explicitly disclose of a *trash register control signal* which programmably disables writing to said trash register because Qiu does not explicitly disclose of how the value of the counter travels to whatever logic is comparing the value with zero, as seen in step 7 of Figure 8.

On the other hand, Kissell discloses of the concept of a signal (col. 8, lines 17, inhibit/burn signal line).

Rationales for arriving at a conclusion of obviousness suggested by the Supreme Court's decision in KSR include combining prior art elements according to known methods to yield predictable results, and it would have been readily recognized to one of ordinary skill in the art at the time of the invention that *a signal* would be able to be used to carry the value in the counter to whatever logic is comparing the value with zero.

Therefore, a sufficient rationale for arriving at a conclusion of obviousness has been arrived at.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Keith Vicary/

Examiner, Art Unit 2183

Keith Vicary

Conferees:

/Eddie P Chan/

Supervisory Patent Examiner, Art Unit 2183

/Manorama Padmanabhan/

Quality Assurance Specialist, TC 2100, WG 2180